002717.P039 *PATENT*

UNITED STATES PATENT APPLICATION

FOR

LOAD-SHARING TECHNIQUE FOR DISTRIBUTING MULTI-PROTOCOL LABEL SWITCHING PROTOCOL ENCAPSULATED FLOWS ACROSS MULTIPLE PHYSICAL LINKS

INVENTORS:

CEDELL ALEXANDER
LANCE RICHARDSON
OLEN STOKES

PREPARED BY:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1026

(503) 684-6200

EXPRESS MAIL NO. EL546137485US

LOAD-SHARING TECHNIQUE FOR DISTRIBUTING MULTI-PROTOCOL LABEL SWITCHING PROTOCOL ENCAPSULATED FLOWS ACROSS MULTIPLE PHYSICAL LINKS

FIELD OF THE INVENTION

[0001] The invention relates to data networking. More specifically, the invention relates to techniques for load sharing of Multi-Protocol Label Switching (MPLS) Protocol encapsulated data across multiple physical links.

BACKGROUND

[0002] It is common in the networking field for data from a high-speed source to be transferred across a lower-speed link. It may be desirable to distribute flows from the high-speed source across multiple lower-speed physical links in a load-sharing manner. The load sharing allows the multiple lower-speed links to simulate a higher-speed physical link. For example, data flows received from a high-speed packet over SONET (PoS) source may be distributed to multiple lower-speed Ethernet links.

[0003] Mapping of data from the high-speed source to the multiple lower-speed links is typically accomplished using a hashing function. Typically, the Point-to-Point Protocol (PPP) is run over the PoS link. However, mapping is further complicated by the potential for data flows to be transmitted according to different protocols. PPP, for example, can support two commonly used network control protocols known as the Internet Protocol Control Protocol (IPCP), which is used to transport IP packets, and the Bridging Control Protocol (BCP), which is used to transport Layer-2 MAC frames. In order to ensure that all of the packets for a given flow are transmitted on the same physical link to avoid potential problems associated with packet misordering, different hashing functions are used for the different protocols.

[0004] For example, the mapping, or hashing, function for the IP packets might operate on the 5-tuple {source IP address, destination IP address, IP protocol type, source port number, destination port number}, and the mapping function for MAC packets may operate on the pair {destination MAC address, source MAC address}. In a standard PPP environment, the determination of which hashing mechanism to use can be based on the Protocol ID field of the PPP header, which identifies the packet as an IP packet or an encapsulated MAC frame.

[0005] A problem exists when packets arriving from the PoS link are encapsulated according to the MultiProtocol Label Switching (MPLS) protocol. The problem exists because the Protocol ID field of the PPP header contains a value that indicates the packet is MPLS encapsulated, and therefore does not indicate the format/type of the encapsulated packet. On a PPP link, an MPLS encapsulated packet contains a MPLS shim header that immediately follows the PPP header. The shim header contains one or more labels (referred to as the label stack) that are used to switch the packet through the network. The labels may also be used to access a label database to determine the type of data carried by the packet. However, providing access to the label database for each component of the network switch that determines the type of data carried by a packet can be cumbersome and expensive.

[0006] Note that the situation is similar for other types of high-speed sources. For example, when the source is a 10 Gbps Ethernet interface and the lower-speed interfaces are 1 Gbps Ethernet links. In the case of an Ethernet source interface, the EtherType field in the Ethernet frame is analogous to the PPP Protocol ID field. The problem is the same in both cases (i.e., the EtherType field contains a value that indicates the packet is MPLS encapsulated, and therefore does not indicate the format/type of the encapsulated packet), and the solution presented for the PoS source is also directly applicable to an Ethernet source interface.

SUMMARY OF THE INVENTION

[0007] Techniques for load sharing of Multi-Protocol Label Switching (MPLS) Protocol encapsulated data across multiple physical links are described. A protocol format in which a packet is formatted is determined based on one or more label values in a header of a MultiProtocol Label Switching (MPLS) formatted packet, where label values in a first range indicate a first protocol type and label values in a second range indicate a second protocol type. A physical link is selected from a plurality of physical links based on the one or more label values.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings in which like reference numerals refer to similar elements.

Figure 1 illustrates one embodiment of load sharing between two switches.

Figure 2 is a block diagram of one embodiment of a switch that analyzes the header of a MPLS packet to determine the type of data encapsulated by the MPLS packet.

Figure 3 is a flow diagram of one embodiment of a technique for mapping flows in a load sharing environment.

DETAILED DESCRIPTION

[0008] Techniques for load sharing of Multi-Protocol Label Switching (MPLS) Protocol encapsulated data across multiple physical links are described. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the invention. It will be apparent, however, to one skilled in the art that the invention can be practiced without these specific details. In other instances, structures and devices are shown in block diagram form in order to avoid obscuring the invention.

[0009] Reference in the specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the invention. The appearances of the phrase "in one embodiment" in various places in the specification are not necessarily all referring to the same embodiment.

[0010] When Internet Protocol (IP) packets or Layer-2 MAC frames are encapsulated in a MultiProtocol Label Switching (MPLS) packet, one or more labels are assigned to identify the path through the MPLS network on which the packet is to be passed. From this information, the underlying protocol (e.g., IP, MAC) can be inferred. In one embodiment, the value assigned to the one or more labels is selected to identify the underlying protocol. In one embodiment, IP packets have label values in a first range and MAC packets have label values in a second range. For example, IP packet labels can be within the range 0 to 300 while MAC packet labels can be within the range of 301 to 600.

[0011] **Figure 1** illustrates one embodiment of load sharing between two switches. Link 100 is a high-speed link, for example, a packet over SONET (PoS) link. SONET links are high-speed links (i.e., 51.840 Mbps and up) that carry data between switches. Switch 110 is coupled to link

100 to receive data from link 100 and transmit data over links 120 to switch 130. However, the links 120 provide lower individual bandwidth than link 100.

[0012] To implement load sharing and increase the data rate between switch 110 and switch 130, switch 110 can transmit data received from link 100 to switch 130 over multiple links 120. Generally, multiple links 120 are lower-speed links, for example, Ethernet. Because packet ordering can be important, each of the flows within the data stream should be mapped to the same physical link.

[0013] In the simple example of Figure 1, switch 130 receives data over multiple links 120 and transmits the data over high-speed link 140. Other, more complex, data flow routing can also be supported using load sharing techniques. If switch 110 supports the Point-to-Point Protocol (PPP), a mapping function can be performed on the Protocol ID field within the PPP header. The value stored in the Protocol ID field indicates whether the packet is formatted as an Internet Protocol Control Protocol (IPCP) packet to encapsulate Internet Protocol (IP) frames or as a Bridging Control Protocol (BCP) packet to encapsulate Media Access Control (MAC) frames.

[0014] Because IPCP and BCP packets have different header structures, the information used to map IPCP and BCP traffic to physical links is different. In one embodiment, IPCP traffic is mapped using a hash function on the 5-tuple {source IP address, destination IP address, IP protocol type, source port number, destination port number} and BCP traffic is mapped using a hash function on the pair {destination MAC address, source MAC address}. Other hash or mapping functions can also be used. The result of the hash function is used to select a physical link on which the frame is transmitted.

[0015] When MultiProtocol Label Switching (MPLS) is used, the Protocol ID field of the PPP header does not indicate the protocol type of the packet. Therefore, a different mapping technique is required for MPLS traffic. Upon entry to a MPLS network, an additional header is added to the packet. The header includes one or more labels (also referred to as the label stack), which are values that represent one or more paths through the MPLS network.

[0016] MPLS networks include two types of routers: Label Edge Routers (LERs) and Label Switch Routers (LSRs). LERs include ingress LERs where the packet enters the MPLS network and egress LERs where the packet exits the MPLS network. The ingress LER generates the header with the one or more labels. The header is a shim header that follows the PPP header in which the Protocol ID field indicates that the packet is a MPLS packet. The path that the packet follows through the MPLS network is the Label Switched Path (LSP).

[0017] In order to infer the type of packet encapsulated within the MPLS packet, the values of the labels are partitioned into multiple ranges, each of which indicates a type of data protocol. In one embodiment, the ranges are: 0-299 for IP LER labels; 300-599 for MAC LER labels; and over 599 for LSR labels. In alternate embodiments, other ranges and/or a different number of ranges can be used. Because LSPs are unidirectional and each router selects the label values assigned to the LSP segments from which the routers receive packets, using different ranges for different types of packets can allow a LER to determine the type of packet that is received by analyzing the labels in the MPLS header. The embodiment illustrated in **Figure 1** is readily extensible to a configuration where switch 110 is an I/O module installed in switch 130, and links 120 are connections to the backplane of switch 130.

[0018] **Figure 2** is a block diagram of one embodiment of a switch that analyzes the header of a MPLS packet to determine the type of data encapsulated by the MPLS packet. Components of a switch other than those used to determine the type of data encapsulated by the MPLS packet have been omitted for reasons of simplicity.

[0019] Line interface 200 is coupled to receive data from link 100. Line interface 200 can be, for example, an optical to electrical converter than converts optical signals to electrical signals. Any type of line interface known in the art can be used. Input buffer 210 is coupled to receive data from line interface 200. Input buffer temporarily stores data prior to processing by other components of the switch.

[0020] MPLS header interpreter 220 is coupled to input buffer 210 to read the MPLS header and determine the label values within the MPLS header. MPLS header interpreter 220 also analyzes the label values to determine the type of packet encapsulated within the MPLS packet. MPLS header interpreter 220 can be implemented as hardware, software, or any combination of hardware and software. In one embodiment, MPLS header interpreter 220 processes the label values according to the following pseudo-code, which is described in greater detail below.

```
    if (one label in MPLS shim header)
    {
        if ((label is in IP Label Partition) || (label == 0))
            use IP hash on packet following label
        else if (label is in MAC Label Partition)
            use layer-2 hash on packet following label
        else // label is in LSR Label Partition
            use hash based on label
    }
else // multiple labels in stack
{
    if (first label in stack is in IP Label Partition)
    {
        if (second label in stack is in MAC Label Partition && is last label in stack)
            use layer-2 hash on packet following label stack
        else if (second label in stack is in IP Label Partition && is last label in stack)
            use IP hash on packet following label stack
        else
            use hash based on all labels in stack
        }
else
        use hash based on all labels in stack
}
```

[0021] Mapping function services 230 performs the mapping function based on the result of the analysis performed by MPLS header interpreter 220. In one embodiment, MPLS header interpreter 220 sends a signal to mapping function services 230 indicating which mapping function (or hash function) to use with the associated packet. Based on the function to be used, mapping function services 230 can retrieve the appropriate header information to compute the mapping function. Mapping function services can be implemented as software, hardware or any combination of hardware and software.

[0022] Packet interpreter 250 is coupled to input buffer 210 and receives the packet and packet header from input buffer 210. Packet interpreter 250 provides header information to mapping function services 230 for use in performing the mapping functions. Mapping function services 230 provides packet interpreter 250 with the result of the mapping function. Packet interpreter 250 directs the packet to one of a set of output buffers (e.g., 240, 242, 244) based on the result of the mapping function. The output buffer transmits the packet across one of links 120.

[0023] In one embodiment, mapping function services 230 performs hash function on the 5-tuple {source IP address, destination IP address, IP protocol type, source port number, destination port number} for IP traffic; however, other mapping/hash values can also be used. In one embodiment, mapping function services 230 performs a hash function on the pair {destination MAC address, source MAC address} for MAC traffic; however, other mapping/hash values can also be used.

[0024] **Figure 3** is a flow diagram of one embodiment of a technique for mapping flows in a load sharing environment. A MPLS packet is received at 300. The MPLS packet includes headers according to multiple protocols. The MPLS header including the label stack is located at 305. The MPLS header interpreter determines whether the stack includes multiple labels at 310.

[0025] If the stack does not include multiple labels, the MPLS header interpreter determines whether the label value is within the IP LER label range at 320. The IP LER label range is a set of predetermined label values used to indicate that a packet is an IP packet. If the label is within the IP LER label range, the packet is mapped using the IP function at 325.

[0026] If the label value is not within the IP LER label range at 320, the MPLS header interpreter determines whether the label value is within the MAC LER label range at 330. If the label value is within the MAC LER label range at 330, the packet is mapped using the MAC function at 335. Otherwise, the packet is mapped based on the label value at 340.

[0027] By analyzing the label stack to determine the type of packet that is encapsulated in a MPLS packet, the MPLS header interpreter can perform a function that would otherwise require a label database. Thus, by storing a small number of values representing the relevant ranges rather than the full database, network switches can be made less complex and less expensive.

[0028] If, at 310, the MPLS header interpreter determines that the stack includes multiple levels, the MPLS header interpreter determines whether the first label is in the IP label range at 350. If the first label is not in the IP label range, the packet is mapped using the label values at 355.

[0029] If the first label is in the IP label range, the MPLS header interpreter determines whether the second label in the stack is in the MAC range and is the last label in the stack at 360.

If the second label is in the MAC range and is the last label in the stack, the packet is mapped using the MAC function at 365.

[0030] If the second label is not in the MAC range or is not the last label in the stack, the MPLS header interpreter determines whether the second label in the stack is in the IP range and is the last label in the stack at 370; if so, the packet is mapped using the IP function at 375. If the second label in the stack is not in the MAC range, is not in the IP range, or is not the last label in the stack, the packet is mapped based on the label values at 380.

[0031] In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes can be made thereto without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.